

BAB 2

TINJAUAN PUSTAKA

2.1 Berita Hoax

Hoax adalah informasi yang sengaja direkayasa untuk menutupi fakta sebenarnya. Dalam istilah lain, hoax dapat diartikan sebagai upaya memutarbalikkan fakta dengan menyajikan informasi yang tampak meyakinkan, namun tidak dapat dibuktikan kebenarannya. Hoax juga sering digunakan untuk mengaburkan fakta dengan cara membanjiri media dengan informasi palsu, sehingga kebenaran sulit ditemukan di tengah arus informasi yang tidak valid [15]. Menurut laporan Kementerian Komunikasi dan Informatika (Kominfo), pada tahun 2023 saja tercatat lebih dari 1.600 hoax yang tersebar luas di Indonesia, dengan isu-isu utama terkait politik dan kesehatan [16].

Berita hoax memiliki beberapa karakteristik khusus yang memudahkan penyebarannya. Menurut penelitian Iswara, A.A. [17], karakteristik tersebut mencakup penggunaan akronim, singkatan, serta simbol atau angka yang dapat menyesatkan pembaca. Selain itu, hoax sering memanfaatkan ejaan dan tanda baca yang tidak konvensional untuk menarik perhatian, serta menggunakan emotikon atau gambar yang memberikan kesan emosional untuk mendukung klaim yang salah. Fungsi utama berita hoax adalah untuk memanipulasi fakta dan membangun daya persuasif melalui tindak tutur *assertive*, *expressive*, dan *directive*, sehingga pembaca mudah terpengaruh dan mempercayai informasi yang tidak benar. Karakteristik dan tujuan ini menunjukkan tantangan signifikan dalam pengembangan sistem deteksi otomatis, karena berita hoax sering kali dirancang untuk menyerupai berita fakta. Oleh karena itu, penelitian ini berfokus pada pengembangan metode yang dapat mengenali pola-pola teks tersebut dengan menggunakan teknik ekstraksi fitur seperti *CountVectorizer* dan *TF-IDF Vectorizer*.

2.2 Feature Extraction

Feature extraction adalah proses penting dalam pengolahan data yang bertujuan untuk merepresentasikan data yang tidak terstruktur ke dalam bentuk yang dapat diproses oleh algoritma pembelajaran mesin [18, 19]. Dalam klasifikasi teks, feature extraction memungkinkan informasi penting dalam dokumen dipertahankan, sementara dimensi data dikurangi untuk meningkatkan efisiensi dan akurasi model. Proses ini menjadi langkah utama dalam berbagai aplikasi, termasuk analisis sentimen, klasifikasi teks, dan deteksi berita hoax.

2.2.1 Count Vectorizer

CountVectorizer adalah teknik ekstraksi fitur yang mengubah teks menjadi representasi numerik berdasarkan frekuensi kemunculan kata dalam dokumen [20]. Metode ini sederhana dan efektif dalam merepresentasikan teks yang tidak terstruktur. Misalnya, pada teks 'Berita Viral Hari Ini' dan 'Berita Terkini Indonesia', Count Vectorizer menghasilkan representasi sebagai berikut:

Tabel 2.1. Matriks Frekuensi Kata dari CountVectorizer

Dokumen	Berita	Viral	Hari	Ini	Terkini	Indonesia
Dokumen 1	1	1	1	1	0	0
Dokumen 2	1	0	0	0	1	1

Tabel 2.1 menunjukkan hasil representasi numerik teks menggunakan CountVectorizer, di mana setiap kolom merepresentasikan kata unik dalam kumpulan dokumen, dan setiap nilai dalam tabel menunjukkan frekuensi kemunculan kata tersebut dalam dokumen tertentu. Sebagai contoh, pada "Dokumen 1," kata "Berita," "Viral," "Hari," dan "Ini" masing-masing muncul satu kali, sedangkan kata "Terkini" dan "Indonesia" tidak muncul sama sekali. Sementara itu, pada "Dokumen 2," kata "Berita," "Terkini," dan "Indonesia" masing-masing muncul satu kali, sementara kata-kata lainnya tidak muncul.

Representasi ini memungkinkan teks yang awalnya tidak terstruktur untuk diubah ke dalam bentuk numerik yang dapat diproses lebih lanjut oleh

algoritma pembelajaran mesin. Dengan cara ini, CountVectorizer menjadi salah satu teknik fundamental dalam pemrosesan teks untuk berbagai aplikasi, termasuk klasifikasi teks dan analisis sentimen.

2.2.2 TF-IDF Vectorizer

TF-IDF (Term Frequency-Inverse Document Frequency) Vectorizer adalah teknik ekstraksi fitur yang menggabungkan dua metrik utama, yaitu Term Frequency (TF) dan Inverse Document Frequency (IDF), untuk memberikan bobot pada setiap kata dalam dokumen [21]. Teknik ini memberikan bobot yang lebih tinggi pada kata-kata yang jarang muncul di seluruh dokumen, tetapi memiliki makna yang signifikan dalam dokumen tertentu. Dengan pendekatan ini, TF-IDF Vectorizer tidak hanya mampu merepresentasikan teks dalam bentuk numerik, tetapi juga menonjolkan kata-kata yang lebih relevan dalam konteks analisis. Pendekatan ini telah terbukti meningkatkan akurasi dan efektivitas dalam berbagai aplikasi pengolahan teks, seperti klasifikasi teks, analisis sentimen, dan deteksi berita hoax [22].

TF mengukur seberapa sering suatu kata muncul dalam sebuah dokumen relatif terhadap panjang dokumen tersebut. Rumus TF adalah:

$$TF(t, d) = \frac{f(t, d)}{\text{Jumlah total kata dalam dokumen } d} \quad (2.1)$$

di mana :

- $f(t, d)$: frekuensi kemunculan kata t dalam dokumen d .
- Pembilang dihitung terhadap total jumlah kata dalam dokumen.

Sebagai contoh, jika kata "hoax" muncul 3 kali dalam sebuah dokumen dengan total 100 kata, maka nilai TF untuk kata "hoax" adalah:

$$TF(\text{hoax}, d) = \frac{3}{100} = 0.03 \quad (2.2)$$

IDF memberikan bobot lebih tinggi pada kata-kata yang jarang muncul di seluruh dokumen dalam kumpulan data (corpus). Formula dasar IDF adalah:

$$IDF(t) = \text{Log} \frac{N}{n(t)} \quad (2.3)$$

di mana :

- N : jumlah total dokumen dalam kumpulan data.
- $n(t)$: jumlah dokumen yang mengandung kata t .

Jika sebuah kata muncul di banyak dokumen, nilai IDF akan lebih kecil, karena kata tersebut dianggap kurang signifikan. Sebagai contoh, jika kata "berita" muncul dalam 50 dari 100 dokumen, maka nilai IDF -nya adalah:

$$IDF(\text{berita}) = \text{Log} \frac{100}{50} = \log 2 \approx 0.301 \quad (2.4)$$

Sebaliknya, jika kata "hoax" hanya muncul di 5 dokumen, nilai IDF -nya akan lebih besar:

$$IDF(\text{hoax}) = \text{Log} \frac{100}{5} = \log 20 \approx 1.301 \quad (2.5)$$

Dengan menggabungkan kedua metrik ini, TF-IDF memberikan bobot lebih tinggi pada kata-kata yang jarang muncul namun memiliki makna penting dalam konteks dokumen tertentu. Pendekatan ini sangat efektif untuk mendeteksi kata-kata yang relevan dalam klasifikasi teks, analisis sentimen, dan tugas lain dalam pemrosesan bahasa alami (*Natural Language Processing*).

2.2.3 N-gram

N-gram adalah teknik ekstraksi fitur yang merepresentasikan teks sebagai urutan elemen berurutan, baik berupa kata maupun karakter [23]. "N" pada N-gram menunjukkan jumlah elemen dalam grup tersebut, seperti unigram (N=1), bigram (N=2), atau trigram (N=3). Teknik ini digunakan untuk menangkap pola atau konteks dalam teks yang tidak dapat dijelaskan dengan pendekatan berbasis kata individu. Sebagai contoh, pada teks "Deteksi hoax semakin penting", hasil N-gram adalah sebagai berikut:

- Unigram (N=1): 'Deteksi', 'hoax', 'semakin', 'penting'
- Bigram (N=2): 'Deteksi hoax', 'hoax semakin', 'semakin penting'
- Trigram (N=3): 'Deteksi hoax semakin', 'hoax semakin penting'

- Unigram & Bigram (N=1, N=2) = 'Deteksi', 'hoax', 'semakin', 'penting', 'Deteksi hoax', 'hoax semakin', 'semakin penting'
- Bigram & Trigram (N=2, N=3) = 'Deteksi hoax', 'hoax semakin', 'semakin penting', 'Deteksi hoax semakin', 'hoax semakin penting'

Pendekatan ini memungkinkan algoritma untuk menangkap pola dan hubungan antar kata, seperti "hoax semakin penting," yang tidak dapat ditangkap hanya dengan unigram. Representasi N-gram sering digunakan dalam berbagai aplikasi, termasuk klasifikasi teks, analisis sentimen, dan prediksi teks [24].

2.3 Naive Bayes

Naive Bayes merupakan algoritma pembelajaran berbasis supervised learning yang digunakan untuk tugas klasifikasi dalam machine learning. Algoritma ini didasarkan pada teorema Bayes, yang menghitung probabilitas suatu hipotesis berdasarkan data yang diamati. Prinsip dasarnya adalah:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

(2.6)

1. $P(h)$ adalah probabilitas awal atau prior bahwa hipotesis h benar.
2. $P(D)$ adalah probabilitas prior dari data D .
3. $P(D | h)$ adalah probabilitas data D terjadi jika hipotesis h benar.
4. $P(h | D)$ adalah probabilitas posterior bahwa hipotesis h benar berdasarkan data D .

Naive Bayes bekerja dengan prinsip bahwa setiap fitur dalam data dihitung secara terpisah untuk menentukan probabilitas kelas tertentu. Pendekatan ini membuat perhitungan probabilitas menjadi efisien dan sederhana. Dalam tugas klasifikasi teks, algoritma ini sering digunakan bersama dengan teknik ekstraksi fitur seperti CountVectorizer atau TF-IDF Vectorizer, yang merepresentasikan data teks dalam bentuk numerik.

Varian Multinomial Naive Bayes merupakan salah satu varian yang paling sering digunakan untuk data berbasis teks [25]. Varian ini dirancang untuk bekerja dengan data yang direpresentasikan dalam bentuk frekuensi kata, seperti hasil dari teknik ekstraksi fitur. Algoritma ini dapat menangkap pola distribusi kata dalam dokumen, sehingga sangat sesuai untuk tugas-tugas seperti klasifikasi teks dan deteksi berita hoax.

sebagai contoh pada tabel 2.1 terdapat 2 dokumen, dokumen 1 dilabeli sebagai hoax dan dokumen 2 dilabeli dengan fakta. menghitung probabilitas prior untuk kelas hoax ($P(\text{hoax})$) dan kelas fakta ($P(\text{fakta})$) pada persamaan 2.3 dan 2.3.

$$P(\text{Hoax}) = \frac{\text{Jumlah dokumen Hoax}}{\text{Jumlah Total Dokumen}}$$

$$= \frac{1}{2} = 0.5 \quad (2.7)$$

$$P(\text{Fakta}) = \frac{\text{Jumlah dokumen Fakta}}{\text{Jumlah Total Dokumen}}$$

$$= \frac{1}{2} = 0.5 \quad (2.8)$$

menghitung setiap kata dalam dokumen berdasarkan frekuensi kata dalam setiap kelas. sebagai contoh, untuk kata "berita" pada persamaan 2.3 dan 2.3.

$$P('berita' | \text{Hoax}) = \frac{\text{Frekuensi 'berita' pada Hoax} + 1}{\text{Total kata pada Hoax} + \text{Jumlah kata unik}}$$

$$= \frac{1}{4+6} = 0.2 \quad (2.9)$$

$$P('berita' | \text{Fakta}) = \frac{\text{Frekuensi 'berita' pada Fakta} + 1}{\text{Total kata pada Fakta} + \text{Jumlah kata unik}}$$

$$= \frac{1}{3+6} = 0.222 \quad (2.10)$$

serta melakukan perhitungan juga untuk kata "hoax" dan "terpercaya" pada persamaan 2.11 dan 2.12.

$$P('hoax' | \text{hoax}) = \frac{1+1}{4+6} = 0.2, \quad P('hoax' | \text{Fakta}) = \frac{0+1}{3+6} = 0.111 \quad (2.11)$$

$$P(\text{'terpercaya'}|\text{Hoax}) = \frac{0+1}{4+6} = 0.1, \quad P(\text{'terpercaya'}|\text{Fakta}) = \frac{1+1}{3+6} = 0.222 \quad (2.12)$$

Menghitung probabilitas posterior pada hoax dan fakta menggunakan rumus Teorema Bayes pada persamaan 2.13 dan 2.14.

$$\begin{aligned} P(\text{Hoax} | \text{Dokumen}) &= P(\text{Hoax}) \times P(\text{'berita'}|\text{Hoax}) \times \\ &\quad P(\text{'hoax'}|\text{Hoax}) \times P(\text{'terpercaya'}|\text{Hoax}) \\ &= 0.5 \times 0.2 \times 0.2 \times 0.1 \\ &= 0.002 \end{aligned} \quad (2.13)$$

$$\begin{aligned} P(\text{Fakta} | \text{Dokumen}) &= P(\text{Fakta}) \times P(\text{'berita'}|\text{Fakta}) \times \\ &\quad P(\text{'hoax'}|\text{Fakta}) \times P(\text{'terpercaya'}|\text{Fakta}) \\ &= 0.5 \times 0.222 \times 0.111 \times 0.222 \\ &= 0.00274 \end{aligned} \quad (2.14)$$

Membandingkan nilai $P(\text{Fakta}|\text{Dokumen})$ dan $P(\text{Hoax}|\text{Dokumen})$ untuk menentukan label kelas hoax atau valid. maka $P(\text{Fakta}|\text{Dokumen}) > P(\text{Hoax}|\text{Dokumen})$ adalah **Fakta**.

2.4 Evaluation Metrics

Evaluation metrics adalah metrik yang digunakan untuk mengukur kinerja model pembelajaran mesin berdasarkan hasil prediksi yang dihasilkan. Dalam tugas klasifikasi, terutama deteksi berita hoax, metrik ini penting untuk mengevaluasi sejauh mana model dapat memprediksi kelas secara akurat, termasuk dalam kasus data yang tidak seimbang. Beberapa metrik evaluasi yang sering digunakan meliputi *accuracy*, *precision*, *recall*, dan *F1-score*.

2.4.1 Accuracy

Akurasi adalah metrik yang paling sederhana dan sering digunakan untuk mengevaluasi model klasifikasi. Akurasi dihitung dengan membandingkan jumlah prediksi benar terhadap total jumlah data. Rumus akurasi adalah:

$$Accuracy = \frac{\text{Jumlah prediksi benar}}{\text{Total jumlah data}} \quad (2.15)$$

Meskipun akurasi memberikan gambaran umum tentang performa model, metrik ini dapat menjadi kurang informatif jika dataset tidak seimbang, di mana satu kelas memiliki jumlah data yang jauh lebih besar dibandingkan kelas lainnya.

2.4.2 Precision

Precision mengukur sejauh mana prediksi positif yang dihasilkan oleh model benar-benar termasuk dalam kelas positif. Presisi didefinisikan sebagai:

$$Precision = \frac{\text{True Positive (TP)}}{\text{True Positive (TP) + False Positive (FP)}} \quad (2.16)$$

Metrik ini penting dalam konteks deteksi berita hoax, karena memastikan bahwa model tidak memberikan terlalu banyak prediksi positif palsu yang dapat menyesatkan pengguna.

2.4.3 Recall

Recall, juga dikenal sebagai sensitivitas atau true positive rate, mengukur sejauh mana model dapat mendeteksi semua data yang termasuk dalam kelas positif. Recall dirumuskan sebagai:

$$Recall = \frac{\text{True Positive (TP)}}{\text{True Positive (TP) + False Negative (FN)}} \quad (2.17)$$

Metrik ini relevan dalam mendeteksi berita hoax, di mana penting untuk memastikan bahwa semua berita hoax dalam dataset dapat terdeteksi oleh model.

2.4.4 *F1-score*

F1-Score adalah metrik yang menggabungkan presisi dan recall dalam satu nilai tunggal, dengan menghitung rata-rata harmonik keduanya. Rumus *F1-score* adalah:

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.18)$$

F1-score berguna ketika dataset memiliki ketidakseimbangan antara kelas positif dan negatif, karena memberikan gambaran yang lebih seimbang antara presisi dan recall.

